AMENDMENTS TO THE CLAIMS:

This listing of claims will replace all prior versions, and listings, of claims in the application:

LISTING OF CLAIMS:

1. (Currently Amended) A system ~~comprising means~~ for sorting incoming data packets in real time, the system comprising:[[;]]

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time; [[and]]

queue means responsive to said assignment means for storing and arranging ~~queuing~~ said packet records ~~sorted packets~~ for output in said exit order; and

memory means for storing said packets or data portions thereof [[;]] , wherein said packets or data portions being output from the memory means in accordance with the corresponding packet records being output from the queue means.

2. (Currently Amended) A system as claimed in claim 1, wherein the ~~sorting~~ assignment means is responsive to information contained within a packet whereby to determine an exit order number for that packet.

3. (Currently Amended) A system as claimed in claim 1, wherein the ~~sorting~~ assignment means is responsive to information contained in a table whereby to determine an exit order number for that packet.

4. (Currently Amended) A system as claimed in claim 1, wherein the ~~sorting~~ assignment means is responsive to information associated with a data packet stream in which said packet is located whereby to determine an exit order number for that packet.

5. (Currently Amended) A system as claimed in claim 1, wherein said ~~sorting~~ assignment means is adapted to insert ~~sorted packets~~ packet records in said queue

means in exit order.

6. (Previously Presented) A system as claimed in claim 1, wherein said queue means is a single queue.

7. (Original) A system as claimed in claim 6, wherein said single queue provides a plurality of virtual queues.

8. (Currently Amended) A system as claimed in claim 1, ~~further comprising~~ wherein said queue means comprises a queue manager adapted to insert ~~packets~~ packet records into said queue means in exit order.

9. (Currently Amended) A system as claimed in claim 1, further comprising means to drop certain ~~packets~~ packet records before being output from said queue means.

10. (Currently Amended) A system as claimed in claim 1, further comprising means to drop certain ~~packets~~ packet records before being queued in said queue means.

11. (Currently Amended) A system as claimed in claim 1, wherein[[:]] said ~~sorting~~ assignment means comprises a processor. ~~and said queue means process only packet records containing information about said packets, and data portions of said packets are stored for output in accordance with an exit order determined for the corresponding packet record~~

12. (Currently Amended) A system as claimed in claim [[1]] 11, wherein said ~~sorting means~~ processor comprises a parallel processor.

13. (Currently Amended) A system as claimed in claim 12, wherein said parallel processor is an array processor comprising one or more arrays of processor

elements.

14. (Currently Amended) A system as claimed in claim 13 ~~12,~~ wherein said array processor is a SIMD processor.

15. (Currently Amended) A system as claimed in claim 12, further comprising means to provide access for said parallel ~~processors~~ processor to shared state.

16. (Original) A system as claimed in claim 15, further comprising a state engine to control said access to said shared state.

17. (Currently Amended) A system as claimed in claim 11, further comprising tables of information for assigning said exit order to ~~sorting~~ said packets ~~or said packet records~~, wherein said tables are stored locally to ~~each~~ said processor ~~or to each processor element of a parallel processor~~.

18. (Currently Amended) A system as claimed in claim 17, wherein said tables are for the same class of service on ~~each~~ said processor ~~or on each processor element of a parallel processor~~.

19. (Currently Amended) A system as claimed in claim 17, wherein said tables are for ~~different~~ varying classes of service on ~~different processors or on different~~ said processor ~~elements of a parallel processor~~.

20. (Cancelled)

21. (Currently Amended) A system as claimed in claim 1, wherein said ~~sorting~~ assignment means ~~implement~~ implements algorithms for packet scheduling in accordance with predetermined criteria, ~~such as~~ including WFQ, DFR, congestion avoidance (~~eg~~ e.g. WRED) or other prioritisation and sorting.

22. (Currently Amended) A method for sorting incoming data packets in real time, the method comprising the steps of:

~~sorting~~ assigning incoming data ~~the~~ packets [[into]] an exit order in real time from packet records only, each of said packet records containing information about a corresponding packet;

storing and arranging in a queue means packet records for output in said exit order;

storing in memory means said packets or data portions thereof; and

outputting packets or data portions from the memory means in accordance with the corresponding packet record being output from the queue means.

~~queueing said sorted packets for output in said exit order.~~

23. (Currently Amended) A method as claimed in claim 22, wherein the ~~sorting~~ assigning is responsive to information contained within a packet whereby to assign an exit order number for that packet.

24. (Currently Amended) A method as claimed in claim 22, wherein the ~~sorting~~ assigning is responsive to information contained in a table whereby to determine an exit order number for that packet.

25. (Currently Amended) A method as claimed in claim 22, wherein the ~~sorting~~ assigning is responsive to information associated with a data packet stream in which said packet is located whereby to determine an exit order number for that packet.

26. (Currently Amended) A method as claimed in claim 22, wherein said ~~packets~~ packet records are inserted into a queue means for output in exit order determined by the means performing the ~~sorting~~ assigning.

27. (Currently Amended) A method as claimed in claim 22, comprising inserting ~~sorted packets~~ said packet records into a queue means for output in exit order under control of a queue manager.

28. (Previously Presented) A method as claimed in claim 26, wherein said queuing is performed using a single output queue.

29. (Original) A method as claimed in claim 28, further comprising providing a plurality of virtual queues by means of a single output queue.

30. (Currently Amended) A method as claimed in claim 22, further comprising dropping certain packets before the corresponding packet records are ~~being~~ output from said queue means.

31. (Currently Amended) A method as claimed in claim 22, further comprising dropping certain packets before the corresponding packet records are ~~being~~ queued in said queue means.

32. (Currently Amended) A method as claimed in claim 22, wherein[[:]]

said ~~sorting~~ assigning is ~~and said queuing operations are~~ performed ~~only on packet records containing information about said packets, said method further comprising:~~

~~storing data portions of said packets in said memory for output in accordance with an exit order number determined for the corresponding packet record~~ by a processor.

33. (Currently Amended) A method as claimed in claim ~~22~~ 32, wherein said ~~sorting is performed by~~ processor is a parallel processor.

34. (Currently Amended) A method as claimed in claim 33, wherein said parallel processor is an array processor comprising one or more arrays of processor elements.

35. (Currently Amended) A method as claimed in claim ~~33~~ 34, wherein said

array processor is a SIMD processor.

36. (Currently Amended) A method as claimed in claim ~~33~~ 32, further comprising providing access for said ~~processors~~ processor to shared state under control of a state engine.

37. (Currently Amended) A method as claimed in claim ~~36~~ 32, further comprising providing tables of information for ~~sorting~~ assigning said packets~~ or said packet records~~, wherein said tables are stored locally to ~~each~~ said processor elements~~ or to each processor element of a parallel processor~~.

38. (Currently Amended) A method as claimed in claim 37, wherein said tables are for the same class of service on ~~each~~ said processor elements~~ or on each processor element of a parallel processor~~.

39. (Currently Amended) A method as claimed in claim 37, wherein said tables are for ~~different~~ varying classes of service on ~~different~~ said ~~processors or on each processor element of a parallel~~ processor element.

Claims 40 – 41 (Cancelled)

42. (Currently Amended) A computer system, comprising a data handling system for sorting incoming data packets in real time, the computer system comprising:[[;]]

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time; ~~and~~

queue means responsive to said assignment means for storing and arranging ~~queuing~~ said ~~sorted packets~~ packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the corresponding packet records being output from the queue means.

43. (Currently Amended) A network processing system, comprising a data handling system for sorting incoming data packets in real time, the data handling system comprising;

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time; [[and]]

queue means responsive to said assignment means for storing and arranging ~~queuing~~ said ~~sorted packets~~ packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the corresponding packet records being output from the queue means.

44. (Currently Amended) A computer system adapted to perform sorting of incoming data packets in real time in which said data packets are ~~sorted~~ assigned ~~into~~ an exit order and ~~are queued for~~ output in said exit order, the computer system comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

queue means responsive to said assignment means for storing and arranging said packet records for output in said exit order; and

memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the corresponding packet records being output from the queue means.

45. (Currently Amended) A network processing system adapted to perform sorting of incoming data packets in real time in which said data packets are assigned ~~sorted into~~ an exit order and ~~are queued for~~ output in said exit order, the network processing system comprising:

assignment means, operable only on packet records containing information about said packets, for assigning an exit order to said packets in real time;

queue means responsive to said assignment means for storing and arranging said packet records for output in said exit order; and memory means for storing said packets or data portions thereof;

said packets or data portions being output from the memory means in accordance with the corresponding packet records being output from the queue means.

46. (Original) A computer system as claimed in claim 42 implemented as one or more silicon integrated circuits.

47. (Currently Amended) A computer-readable medium containing instructions which, when executed on a computer processor or processor element, ~~data carrier containing program means adapted to~~ perform sorting of incoming data packets in real time in which said data packets are assigned ~~sorted into~~ an exit order and are ~~queued for~~ output in said exit order, and performs the steps of:

assigning incoming data packets an exit order in real time from packet records only, each of said packet records containing information about a corresponding packet;

storing and arranging in a queue means packet records for output in said assigned exit order;

storing in memory means said packets or data portions thereof; and outputting packets or data portions from the memory means in accordance with the corresponding packet record being output from the queue means.

48. (New) A system as claimed in claim 13, further comprising tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor element of said parallel processor.

49. (New) A system as claimed in claim 48, wherein said tables are for the same class of service on each processor element of said parallel processor.

50. (New) A system as claimed in claim 48, wherein said tables are for varying classes of service on different processor elements of said parallel processor.

51. (New) A system as claimed in claim 48, wherein said processor elements share information from their respective tables, such that:

(a) the information held in the table in one processor element is accessible by other processing element(s) of said parallel processor; and

(b) processor elements have access to other processor elements in said parallel processor, whereby processor elements can perform table lookups on behalf of other processor elements of said parallel processor.

52. (New) A method as claimed in claim 34, further comprising providing tables of information for sorting said packets or said packet records, wherein said tables are stored locally to each processor element of said parallel processor.

53. (New) A method as claimed in claim 52, wherein said tables are for the same class of service on each processor element of said parallel processor.

54. (New) A method as claimed in claim 52, wherein said tables are for varying classes of service on each processor element of a parallel processor.

55. (New) A method as claimed in claim 52, wherein said processor elements share information from their respective tables, such that:

(a) the information held in the table for one processor element is made directly accessible to other processor element(s) of said parallel processor; and

(b) access is provided for said processor elements to tables in other processor elements, whereby processor elements can perform table lookups on behalf of other processor elements of said parallel processor.